
Docker のコンテナ間の 通信について

ヤス

Dockerとは

- コンテナ型の仮想環境構築ツール
 - 従来のゲストOS上に仮想環境を構築するのではなく、ホストマシンのカーネルを利用して、プロセス等を隔離し、別のマシンが動いているかのように動かす
 - ホストマシン上で動作するため、作成したコンテナ間の連携が取りやすい

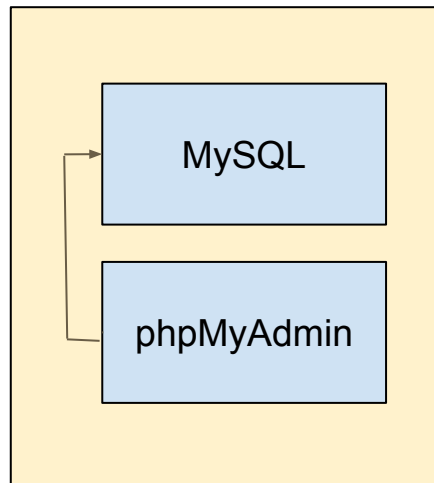
コンテナ間の通信について

- docker-compose を利用
 - Docker をコンテナの操作を簡単にできるツール
 - docker-compose.yml でコンテナの設定をできる
- depends_on の使用することでコンテナ間の通信が可
 - version 2から
 - version 1では link が必要

コンテナ間の通信について

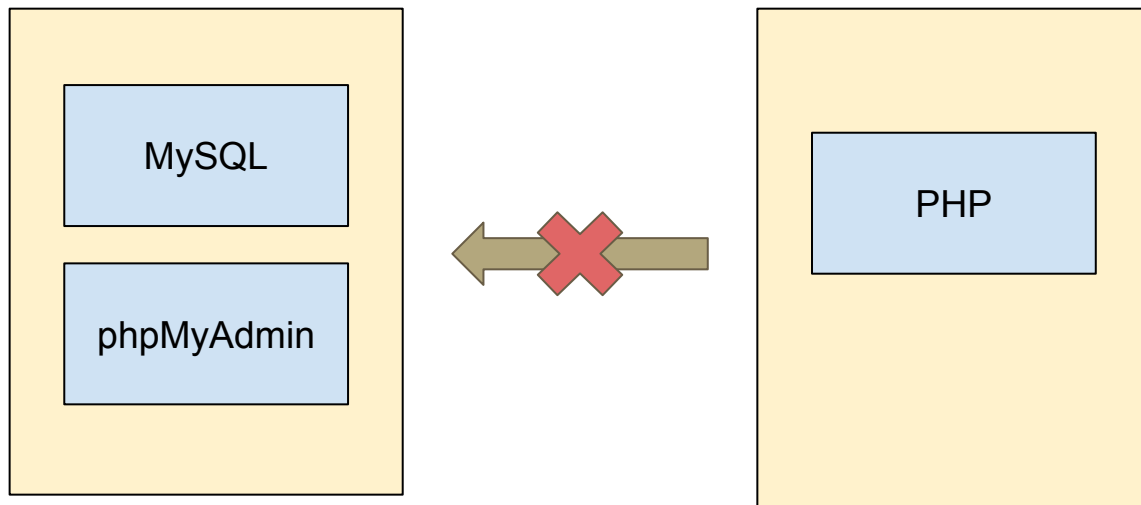
```
version: '2'
services:
  db:
    container_name: "mysql"
    image: mysql:latest
    ports:
      - "3306:3306"

  phpmyadmin:
    container_name: "phpmyadmin"
    image: phpmyadmin/phpmyadmin:latest
    ports:
      - "80:80"
    depends_on:
      - db
```



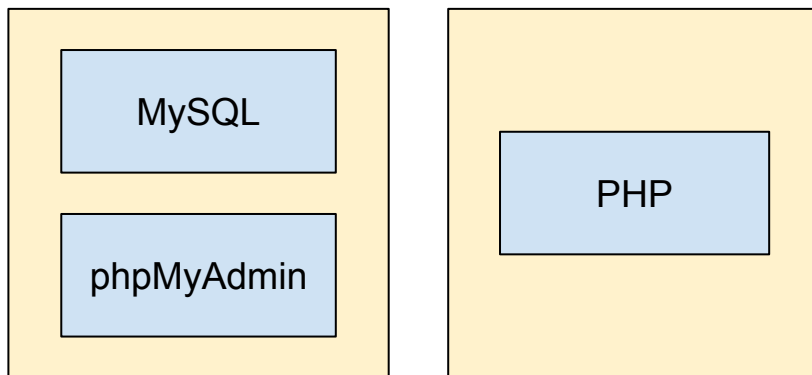
コンテナ間の通信について

- `docker-compose.yml` で定義されていないコンテナへの通信はできない



コンテナ間の通信について

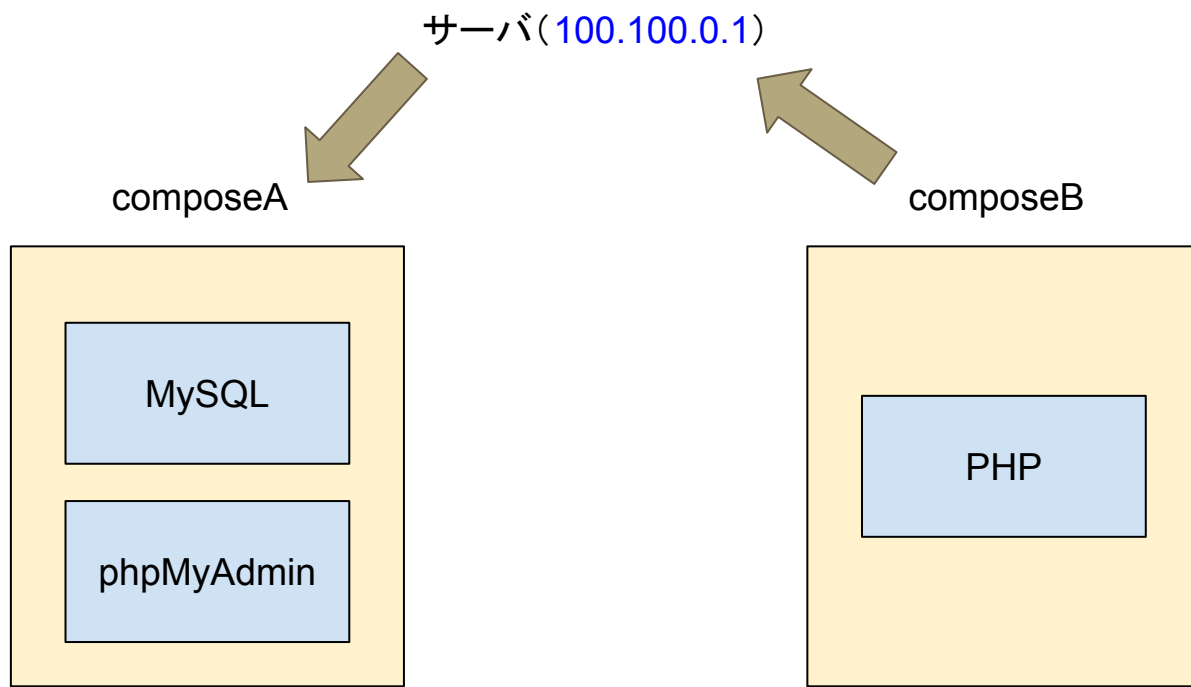
- docker-compose.yml で定義されたコンテナは起動や停止のタイミングが同期するため、目的ごとに分けたい
 - DB用
 - PHP用



コンテナ間の通信について

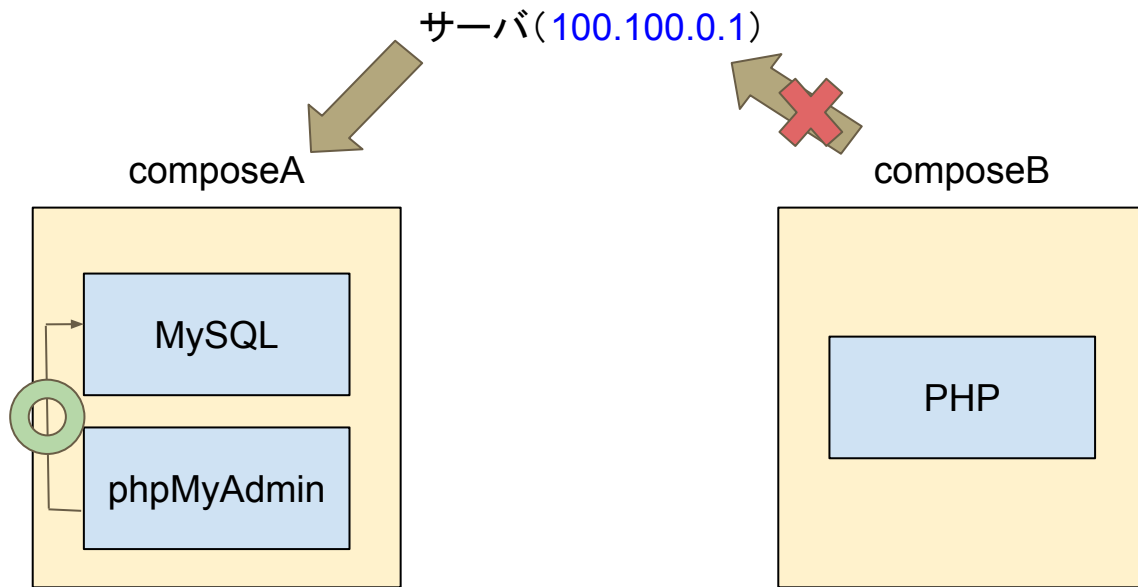
- docker0 (docker のデフォルトネットワーク) を利用
 - docker のコンテナからホスト OS に通信できる
 - localhost のようなもの
 - デフォルトの IP アドレスは動的
 - OS の再起動時等に変更される
 - 設定ファイルを用意することで固定 IP にできる
 - /etc/docker/daemon.js

```
[  
  "bip": "100.100.0.1/24"  
]
```



ファイアウォールを導入

- ファイアウォールを起動すると docker 間の通信も制限される
 - depends_on している通信は可能



ファイアウォールを導入

- コンテナ単位に固定IPを設定する必要がある

```
version: '2'
services:
  php:
    container_name: php
    image: php
    networks:
      php_network:
        ipv4_address: 100.100.10.2

networks:
  php_network:
    driver: bridge
    ipam:
      driver: default
      config:
        -
          subnet: 100.100.10.0/24
```

ファイアウォールを導入

- ファイアウォールで通信を許可すると通信可能

